UNIVERSIDAD COMPLUTENSE MADRID

Identificación de patrones de glucemia por tramos de cuatro horas en diabéticos tipo I mediante monitorización continua de glucosa y técnicas estadísticas

Autores:

Miguel Fuentes López y Lucas Segarra Fernández

Director y codirector:Ignacio Hidalgo y Manuel Velasco

Trabajo de fin de grado del Grado en Ingeniería Informática, Facultad de Informática

Términos: Diabetes Mellitus, Agrupamiento, Monitorización continua de glucosa, Patrones en tramos de cuatro horas, Clustering sobre diabeticos, Servidor web en Python, Autenticación delegada

Terms: Diabetes Mellitus, Clustering, Continuous glucose monitorization, Four-hour-slot patterns, Clustering over diabetics, Python web server, Delegated authentication

Resumen

La diabetes es una enfermedad que afecta a la secreción o al efecto de insulina que es la hormona encargada del transporte de glucosa en sangre. Por ello, las personas que la sufren tienen que estar muy pendientes de sus niveles de azúcar y controlar de manera rigurosa su alimentación y actividad física.

Con la evolución de las tecnologías se han desarrollado dispositivos, como el FreeStyle Libre, que permiten la monitorización continua de estos niveles. Con el auge de las técnicas de aprendizaje automático, parece el momento justo de emplear la capacidad de estudiar grandes cantidades de datos en los registros de mediciones que producen estos dispositivos. Hasta la fecha la mayor parte de los trabajos en este campo corresponden a estudios sobre poblaciones.

El objetivo de este trabajo es desarrollar una herramienta informática que analíce patrones que presentan individuos, estudiando los datos en tramos de cuatro horas mediante técnicas de clustering, especialmente no jerárquico pero no solo ello, desarrollando en Python con el módulo sklearn.

Abstract

Diabetes is an illness that impacts on insulin secretion or on its effects. Insulin is the hormone that carries glucose in blood. So diabetics must be extremely aware of their sugar levels and rigorously control both diet and physical activity.

Modern devices, such as FreeStyle Libre, allow continuous monitoring of those levels. Due to automatic learning techniques boom, it might be the moment to boost the ability of studying huge amounts of data on measurements registered by these devices. Up to now, the majority of the work developed in this area consist of studies about populations.

The goal of this project is to develop a software tool that analyze patterns presented by people, grouping data into four hours slots by applying clustering methods, specially, but not only, non hierarchical ones. They were developed in Python with the module sklearn.

Índice

1. Introducción

La Diabetes se define como un conjunto de alteraciones metabólicas caracterizadas por la hiperglucemia (concentraciones elevadas de glucosa en sangre) crónica y trastornos en el metabolismo de los hidratos de carbono, las grasas y las proteínas, como consecuencia de defectos en la secreción de insulina, a la acción de ésta, o a ambas.

Se estima que el 13,8% de la población española vive con esta enfermedad.

La mayoría de los alimentos que ingiere nuestro organismo los metaboliza la glucosa, que es el combustible más utilizado por las células para obtener energía.

El nivel de glucosa, se mantiene constante gracias a la acción de la insulina, que participa en su transformación aportando la energía que necesitan las células.

La insulina es una hormona que produce el páncreas. Su principal efecto es permitir a la glucosa entrar a las células para que éstas, la utilicen generando energía. Los diabéticos padecen de alteraciones en la producción de insulina.

Las complicaciones de la diabetes, se clasifican en agudas, descontroladas y crónicas.

Pueden dar lugar a lesiones en el vaso arterial, originando insuficiencias en el riego sanguíneo de extremidades, pérdida de visión, fallos renales, complicaciones coronarias, pérdida de las extremidades inferiores, etc.

Todo ello supone una carga tanto para el paciente como para el sistema de salud público.

Esta carga es mayor cuando se alcanzan complicaciones, que suelen implicar la hospitalización del enfermo.

La estrategia a seguir para las enfermedades crónicas en general, y también para la diabetes, consiste en la prevención de episodios agudos.

Para ello resulta imprescindible monitorizar de forma contínua a los pacientes. Entre los indicadores empleados para medir la diabetes se encuentran los siguientes, que el paciente puede proporcionar por sí mismo.

Indicador	Frecuencia (veces al día)
Nivel de glucosa en sangre	6
Líquido bebido	1
Peso	1
Temperatura	2 (mañana y tarde)
Medicación tomada para la diabetes	1
Nivel de cetona en la orina	Cada vez que se orine
Cantidad prevista de comida a ingerir	En cada comida
Previsión de ejercicio físico a realizar	Cada vez que vaya a hacer deporte

Tabla 1:Principales indicadores de la diabetes

Actualmente, no existe curación de esta enfermedad. La insuficiencia de concentración de insulina en la sangre, solo se puede corregir mediante la aportación de insulina, inyectandola en la grasa que está debajo de la piel.

Por otra parte, la aplicación de una dosis excesiva de insulina puede provocar hipoglucemia. La hipoglucemia es más temida por sus consecuencias inmediatas: somnolencia, taquicardia, confusión mental, pérdida de conciencia, convulsiones y coma. Estas situaciones, pueden suponer secuelas graves en el organismo del paciente.

Las tecnologías contribuyen de forma importante a:

- optimizar la monitorización de los pacientes.
- ajustar las dosis de insulina a aportar al organismo humano a las cantidades óptimas a partir de las mediciones de glucosa obtenidas.

La cantidad de insulina que se ha de administrar un diabético, puede determinarse tomando como referencia un valor medido de la concentración de glucosa en la sangre

insulina(t) = f [glucosa(t)],

o por el contrario puede tenerse en cuenta además una estimación del nivel de glucosa a corto plazo, valorando también otros factores, como por ejemplo el nivel de ingesta de alimentos y de ejercicio físico a realizar

insulina=f [glucosa(t+t)] .

1.1. Objetivos

Tal y como ha sido mencionado la diabetes no tiene curación. El único tratamiento posible es ajustar el nivel de glucosa en la sangre mediante aportación externa de insulina.

Este trabajo se integra en otro de mayor alcance cuya finalidad es la construcción de un sistema de monitorización continua de pacientes.

Utilizando este modelo los pacientes pueden capturar la información proporcionada por un sensor y transmitirla a un servidor Web. El servidor aloja la información en un sistema de ficheros, donde pueden ser objeto de tratamiento y de consulta tanto por el paciente como por el médico.

Mediante la aplicación de algoritmos se puede determinar la dosis más adecuada y realizar búsqueda de patrones en los niveles de glucemia.

El alcance de este trabajo fin de grado es desarrollar soluciones que permitan :

- La transmisión por parte del paciente por Internet mediante una aplicación Web de las mediciones realizadas por un sensor determinado (FreeStyleLibre).
- La recepción y validación de la información transmitida en el servidor.
- El almacenamiento de la misma en un sistema de ficheros.
- La opción dada al paciente de aplicar tres tipos de algoritmos a estos datos para orientarle sobre cuál es la dosis de insulina que debe administrarse.

Hay dos formas de realizar la cuantificación de insulina a aportar al organismo:

- De modo manual: Este sistema de control se denomina de lazo abierto.
- De forma automática: Este sistema se le conoce como de lazo cerrado.

En el sistema de control manual el médico, o en su caso un sistema inteligente, determina la cantidad de insulina a administrar.

La cuantificación se realiza a partir de los indicadores conocidos y que han sido citados en los antecedentes. Pero es el paciente quien finalmente decide la cantidad de insulina que se va a inyectar, pudiendo seguir o no la recomendación realizada por el médico o por el sistema inteligente.

En el sistema automático se dispone de una medición continua del nivel de glucosa en la sangre. Este valor determina la cuantificación también de forma continua de la dosis de insulina a aplicar. A este sistema se le denomina de lazo cerrado, porque la inyección de insulina se hace de forma automática a partir de las mediciones obtenidas, no depende de la voluntad del paciente.

Existen para ello sistemas de infusión subcutánea continua de insulina. Este tipo de tecnologías ha dado origen a una familia de dispositivos denominados páncreas artificial.

Estos dispositivos constan de un sensor que realiza medidas instantáneas del nivel de glucosa en la sangre, e informa a un sistema de control que regula automáticamente la dosis exacta de insulina, que el sistema infusor ha de administrar en cada momento sin requerir la intervención del paciente.
El páncreas artificial consta por tanto de los siguientes componentes:

- sensor/ medidor de glucosa
- algoritmo de determinación de la dosis de insulina
- sistema de infusión de insulina

Estos componentes pueden estar integrados en un único dispositivo o no.

En el segundo supuesto se encuentran los sistemas de Páncreas Artificial Telemático en los que el algoritmo se ejecuta en un sistema remoto. Este sistema tiene la ventaja de que permite tener en cuenta el historial médico del paciente y los valores de otros indicadores distintos al del nivel de glucosa en la sangre.

Entre los monitores/sensores menos intrusivos, se encuentran los del modelo FreeStyleLibre del laboratorio Abott.

Este dispositivo gestiona la monitorización continua y almacena los resultados en un fichero con formato CSV. También muestra los resultados a través de una pequeña pantalla. Se coloca en el antebrazo y por simple presión y contacto con la piel realiza las mediciones.

1.2. Estado del arte

Este apartado ha sido redactado basándonos en otro de un trabajo, al que nos referiremos como “TrabajoDSI”, que tuvimos que hacer junto con otros compañeros en la asignatura de Diseño de Sistemas Interactivos. Tanto ellos, como nuestro profesor de dicha asignatura, eran conocedores de que elegimos la temática de aquel trabajo con la firme intención de utilizarlo como fuente para este apartado. Tales compañeros están mencionados en el apartado de agradecimientos. El resultado final del mismo está enlazado en la bibliografía.

Hacemos un breve análisis de mercado buscando aplicaciones que se comunican con dispositivos similares. De entre todas las existentes, resumimos a continuación las que consideramos más similares a nuestro proyecto y ofrecemos capturas de pantalla de “Libre Link” y “Si Diary Diabetes Managment”.

- Libre Link: Escaneo rápido de los niveles de glucosa. Mostrando además un historial de las últimas 8 horas y una flecha de tendencia que le indica la dirección en la que va su nivel de glucosa, permitiendo compartir sus lecturas con usuarios que tienen la app.

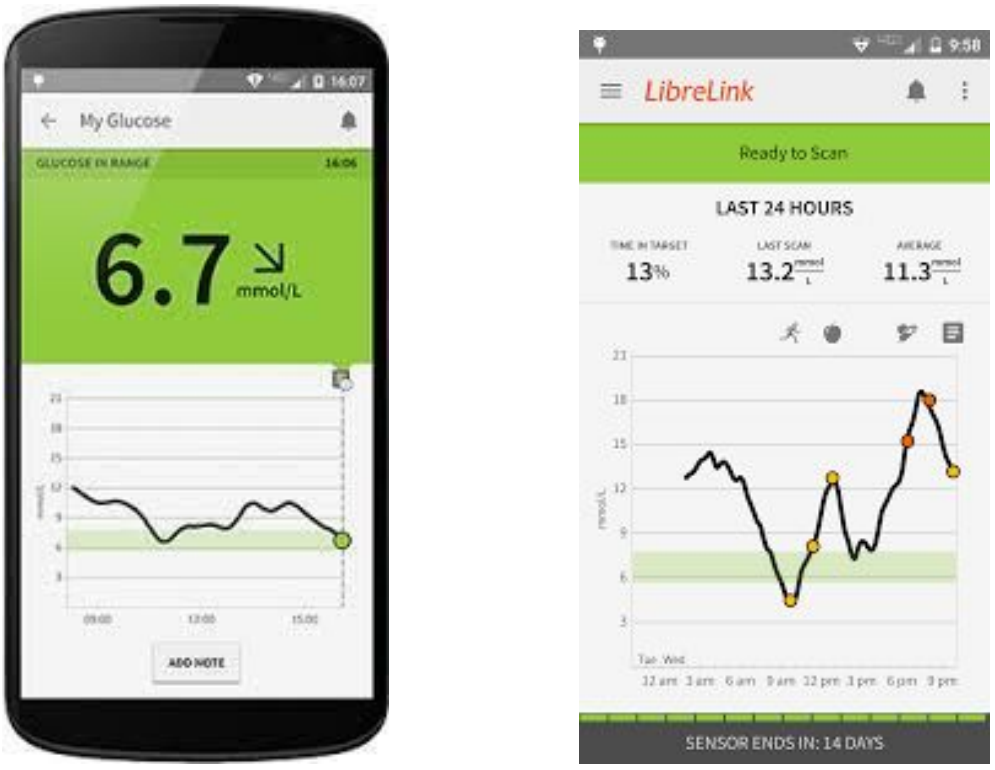

Imagen1:Captura de pantalla de la app “Libre Link”

- OnTrack diabetes: Es una de las aplicaciones relacionadas con la diabetes más descargadas en la PlayStore. La aplicación incluye la posibilidad de crear un widget que permite añadir fácil y rápidamente los niveles de glucosa del usuario.

- FEDE diabetes: La Federación de Diabéticos Españoles (FEDE) creó FEDE diabetes, una aplicación, totalmente gratuita y disponible para iPhone y iPad. La app está dirigida a personas con diabetes (tipo 1 y 2), pero especialmente al público más joven. (Web y iOS).

- SiDiary diabetes management: Un software sencillo para usuarios sin muchos conocimientos de informática y que facilita la comunicación con el médico en caso de incremento drástico.

Imagen2:Captura de pantalla de la app “Libre Link”

- SocialDiabetes: Aplicación para la auto monitorización de la glucemia específicamente diseñada para diabéticos tanto tipo I como tipo II pero más orientado a los primeros .

  En contraposición con estas aplicaciones, la nuestra permite al usuario, asesorado por su doctor preferiblemente, elegir entre tres métodos distintos (Kmeans, Agglomerative Clustering y HDBScan) para aplicar a las mediciones que él mismo nos proporciona.

1.3. Plan de trabajo

Por ser un equipo de dos personas, consideramos más útil repartirnos las tareas, centrándonos cada uno en las propias mientras no fuese más efectivo resolver juntos algún problema concreto que causara un cuello de botella.

Las decisiones de diseño se han tomado de forma consensuada, al igual que las de desarrollo, intentando en el caso de estas últimas además hacerlo de forma previa a comenzar la implementación de cada una.

Durante una breve fase inicial, investigamos y diseñamos a grandes rasgos el flujo de nuestra aplicación.

Al trabajar con algunas tecnologías que apenas conocíamos, no sabíamos de antemano el tiempo que nos iba a llevar cada tarea concreta. Es por ello que una vez decidida la arquitectura de nuestro proyecto, escogimos un modelo de trabajo basado en los planes de desarrollo ágiles. Definiendo tareas genéricas y dividiendo cada una de éstas en tantas (subtareas) como hayan sido necesarias durante la fase de desarrollo.

A partir del momento en el que empezó el desarrollo, para cada hito que nos hemos marcado, hemos iterado estos 3 pasos tantas como ha sido necesario hasta obtener un producto suficientemente similar a lo que buscábamos:

1. Diseño
2. Desarrollo
3. Pruebas

2. Materiales y métodos

Se expone a continuación en qué circunstancias se ha desarrollado este trabajo, así como su organización previa.

2.1. HW disponible

Nos ha sido concedida una CPU localizada en uno de los despachos (el de DACYA), de la segunda planta. También se nos ha comunicado la IP pública asignada a tal host.

Aparte de la CPU, que es nuestro servidor, se nos facilitó un puesto en el mismo despacho donde está emplazada, así como una tarjeta para acceder al mismo.

Para el desarrollo del proyecto, hemos trabajado tanto en dicho despacho, como en cualquier otro sitio mediante ssh.

El dispositivo empleado para obtener las mediciones se denomina FreeStyle Libre Flash Glucose Monitoring System del laboratorio Abbott. Incluye un sensor y un receptor. El tamaño del sensor es el de una moneda de dos euros y contiene un filamento muy fino del diámetro de un pelo (0,4 mm) y de 5 mm de longitud. Consta del citado filamento, electrónica y un chip emisor.

El sensor se adhiere al antebrazo y el filamento se introduce debajo de la piel mediante un aplicador similar a un sello de caucho. Se trata por tanto de un sensor intersticial. Los sistemas de medición intersticial miden la glucosa circulante en los fluidos que hay entre las células de los tejidos bajo la piel y no en el aparato circulatorio (medición capilar). Las mediciones se realizan cada minuto.

Este sensor dispone de una memoria flash capaz de guardar las mediciones de 8 horas. La información se incorpora al receptor cuando se escanea el sensor. El receptor lee los datos cuando está situado entre 1 cm y 4 cm del sensor. Está provisto de una unidad de memoria que permite almacenar las mediciones de 90 días.

Dispone de un micro-puerto USB que permite conexión física con el ordenador del paciente.

2.2. Decisiones de diseño

Determinamos la infraestructura de nuestro proyecto, intentando que el código que generamos esté lo menos acoplado posible. Para ello empleamos el modelo-vista-controlador.

En cuanto a las herramientas empleadas, elegimos Linux y Python por eficiencia, eficacia, ser de software libre, estar relativamente familiarizados con ambas.

Para evitar la gestión de claves, y facilitarnos en cierta medida el cumplimiento de las exigencias de la LOPD, preferimos que los usuarios se identifiquen mediante el sistema de autenticación delegada en Google+.

Este proceso consiste en que sea una tercera parte, en este caso Google+, la que gestione de forma segura, los datos y las claves, y cumpla la legislación pertinente al respecto del almacenamiento de datos.

También para simplificar tareas de seguridad principalmente, hemos considerado más conveniente no utilizar base de datos.

Nuestro único sistema de almacenamiento, es el propio sistema de ficheros de nuestro servidor. En ningún momento se nos ha requerido proporcionar al usuario acceso en línea a los registros de sus ficheros subidos con anterioridad.

Evidentemente el usuario puede conocer, borrar o rectificar cualquiera de los ficheros que se almacenan acerca de él, pero para ello tiene que adjuntarnos por email una fotocopia de su DNI, desde la misma cuenta de correo desde la que subió los ficheros en cuestión.

En cuanto a la información que devolvemos desde la web al usuario, siempre es acerca de un archivo que tiene que subir el propio usuario. De hecho, la aplicación web nunca accede a los ficheros ya existentes. Solo almacena en nuestro sistema de ficheros un archivo cada vez que un usuario lo sube.

2.2.1. Árbol de directorios

Para almacenar el código del proyecto, utilizamos los siguientes directorios:

- main/
    - static/
        - styles.css
    - view/
        - view0.tpl
        - ...
    - controller/
        - webController.py
    - model/
        - meanSerpentForest/
            - src.py
            - PersistenceFile.py

Dónde

- main/ es el directorio raíz del proyecto.
- En static/ almacenamos archivos estáticos, concretamente el archivo de configuración de estilos (CSS).
- En view/ almacenamos las vistas, archivos HTML con extensión .tpl .
- En controller almacenamos el programa encargado de atender las peticiones HTTP.
- En model guardamos la lógica del proyecto.

2.2.2. Tecnologías empleadas

Todo el software que hemos utilizado para realizar el proyecto es libre.

El conjunto de herramientas utilizadas para este trabajo es considerablemente grande. Se listan a continuación las que se consideran esenciales, entendiendo por tales aquellas que necesitamos mencionar para explicar la infraestructura del proyecto.

- A nivel de aplicación hemos decidido utilizar el protocolo HTTPS para la comunicación cliente servidor.
- Para la configuración del servidor, hemos optado por una arquitectura de 4 capas:
    1. El sistema operativo instalado en el servidor es Debian Server.
    2. El servidor web está montado con Python. Mediante la biblioteca Bottle.py se atienden las peticiones de los clientes, y con la biblioteca Cherry.py se escucha el puerto 443.
    3. Como base de datos utilizamos el propio sistema de ficheros del servidor.
    4. Todo módulo de python necesario ha sido mediante la herramienta pip

- La lógica de la aplicación también se ejecuta en Python.

    - La lectura de los ficheros se realiza con el módulo pandas
    - En el tratado de fechas se utilizan las funciones de datetime
    - El tratado de los datos se hace mediante numpy
    - Las técnicas de aprendizaje automático consisten en llamadas a sklearn, aunque también se ha usado durante el desarrollo de la aplicación un módulo llamado HBDSCAN
    - Los gráficos son generados mediante matplotlib
    - El documento pdf final es generado por FPDF
- Para el front-end enviamos al cliente páginas HTML, estas páginas además están estiladas con CSS y dotadas de funcionalidad con JavaScript.
- El cliente accede a nuestra IP, a través de un navegador que le permite enviar y recibir peticiones, que son atendidas por nuestro controlador, recibiendo vistas de éste. Interactuando con estas vistas, podrá ejecutar los métodos de clustering del modelo.
- Para el control de versiones, hemos utilizado Git, en concreto mediante la plataforma de desarrollo colaborativo GitHub.
- Por estar localizado el servidor en la Universidad, y ante la necesidad de realizar modificaciones sobre el código en tiempo real, accederemos al mismo mediante SSH.
- Para permitir al cliente ejecutar programas alojados en el servidor, empleamos CGI.
- Como ya hemos mencionado, el proceso de autenticación se delega en Google+ utilizando para ello
    - OpenID Connect
    - API de Google.

2.2.3. Almacenamiento de la información

A efectos de la almacenar datos, la aplicación permite al usuario subir a nuestro servidor, ficheros CSV (con extensión .txt) que guardamos en una carpeta por medición, anidada en un directorio por fichero recibido, que a su vez está anidado en un directorio por cliente.

Todas las mediciones las almacenamos en un único directorio que llamamos archivos. Este directorio es la raíz del siguiente árbol.

- archivos/
    - usuario1/
        - primera IP
        - subida1/
            - csv.txt
            - informe.pdf
            - img.png
            - ip_de_subida
            - 0/
                - 82
                - 102
                - ...
            - 1/
            - ...
    - usuario2/
    - ..

Dónde:

- csv.txt es el fichero subido por el usuario.
- informe.pdf es el documento de salida.
- img.png es la imagen con los gráficos incluidos en el informe.
- 0,1,2… son las carpetas que agrupan los clusters.
- 82,100… Son los tramos del cluster cuya cifra menos significativa indica el tipo de tramos del día que indican el resto de cifras.

2.2.4. Tareas identificadas

Cabe mencionar, que a pesar de que una buena parte de las mismas las conocíamos desde el momento en el que nos planteamos el proyecto, la mayoría las hemos identificado durante la fase de desarrollo, tal y como suele ocurrir en los procesos de desarrollo ágiles.

Aunque podríamos agrupar las tareas en las que conocíamos previamente. y las que hemos ido identificando sobre la marcha, encontramos más interesante listarlas todas juntas:

1. Servidor
    1. Instalar sistema operativo
    2. Habilitar CGI scripts
    3. Crear scripts de utilidad
2. Web
    1. Atender peticiones HTTP
    2. Definir y maquetar vistas
    3. Interactuar con el modelo
3. Archivos de entrada, clustering y resultados
    3.1. Análisis de la entrada
    3.2. Clustering
        3.2.1 Tipos
        3.2.2 Metricas

3.3. Resultados y representación

4. Seguridad
    1. Autenticación delegada
    2. Identificar y cumplir obligaciones legales
    3. Redactar documento de seguridad
    4. Definir responsables
    5. Adoptar medidas preventivas

2.2.4.1. Estimación de coste

Gran parte del tiempo que hemos dedicado ambos integrantes al proyecto, ha consistido en investigar y probar las tecnologías que hemos utilizado. En ningún momento hemos encontrado conveniente fijarnos plazos medios o largos más allá que entregar el proyecto en junio. Al contrario, definidas las tareas esenciales, el tiempo empleado en cada una de ellas lo han determinado los siguientes factores:

- La dificultad que de por sí tuviese la tarea.
- La importancia que se le hubiera dado a la tarea.
- El nivel de conocimiento del desarrollador sobre las herramientas a emplear.
- La incidencia de la tarea en el proyecto, hasta qué punto afecta a otras tareas.

2.2.4.2. Prioridad

Tal y como caracteriza a los modelos ágiles, cada tarea que hemos asignado, la hemos desarrollado hasta lograr toda la funcionalidad exigida por el producto viable mínimo (MVP) que queríamos obtener.

Evidentemente se ha priorizado por cumplir todos los requisitos de forma eficiente, buscando realizar primero las tareas que pudiesen suponer a posteriori un cuello de botella para el resto del proyecto.

Cómo se ha indicado con anterioridad, el progreso en alguna de las partes del proyecto, ha supuesto en muchas ocasiones, rediseño, ampliación de funcionalidades o cualquier otra modificación en otras tareas.

Una vez obtenido el MVP, se le ha ido añadiendo robustez, funcionalidad y diseño, en ese orden de importancia.

3. Descripción de la aplicación

A continuación se procede a explicar detalladamente tanto el funcionamiento, como el desarrollo de nuestra herramienta.

3.1. Flujo

Se describe a continuación la secuencia que lleva a un cliente a interactuar con nuestra aplicación:

1. El cliente accede a nuestra página de bienvenida, que le ofrece la posibilidad de identificarse con sus credenciales de Google+.
2. Una vez identificado, Google redirige al usuario a nuestra página que permite subir archivos y seleccionar el método de clustering a utilizar, así como, opcionalmente, el número de clusters para los algoritmos no jerárquicos.
3. Si el formato del archivo seleccionado es correcto, se pondrá en marcha la aplicación base que realizará el estudio pertinente.
4. Como consecuencia se generarán unas gráficas y tablas que se reunirán en un archivo en formato PDF, que se mostrará al usuario con las debidas indicaciones.

3.2. Interfaz de usuario

La aplicación web, permitirá al usuario visitar las siguiente páginas:

- landing: Página a la que el usuario puede acceder escribiendo nuestra URL, permite al usuario identificarse con Google+.
- error: Página que se mostrará cada vez que se detecte un fallo.
- upload: Permite al usuario subir ficheros y elegir con qué algoritmo y usando cuantos núcleos quiere que realicemos el clustering.
- show_pdf: Muestra al usuario un documento en formato PDF con los resultados obtenidos.

Mostramos capturas de pantalla obtenidas como cliente (navegador Firefox) de las mismas.

Imagen3:Captura de pantalla de nuestra landing page

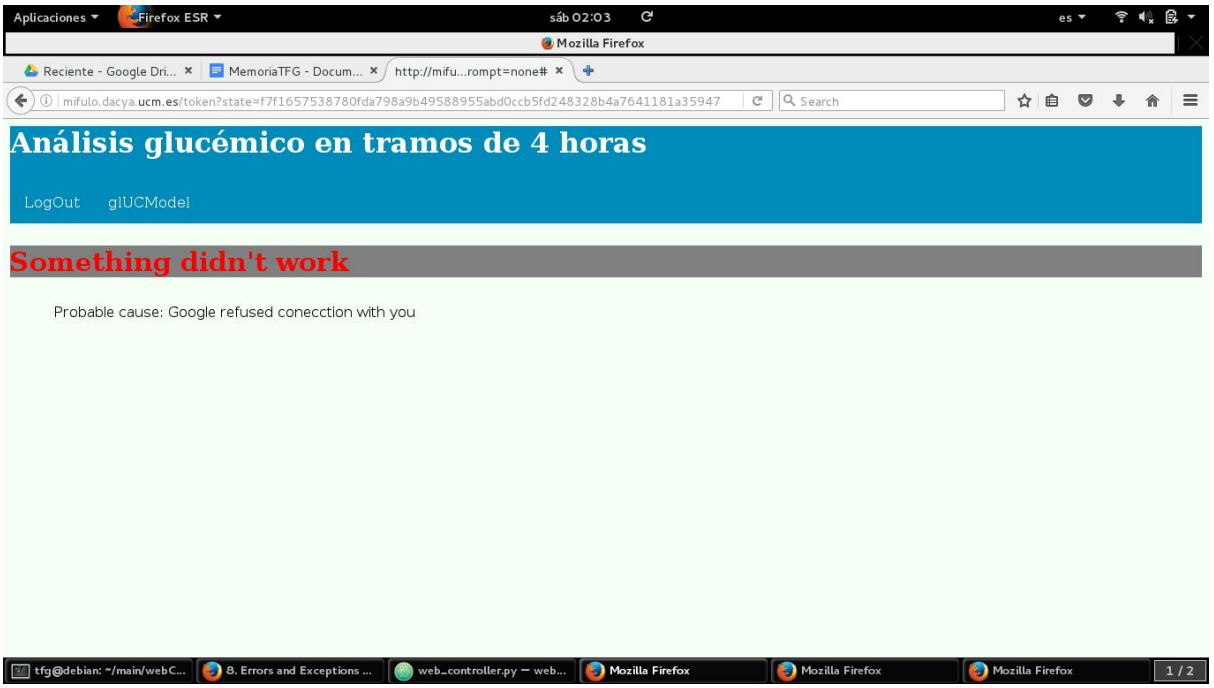

Imagen4:Captura de pantalla de nuestra página de error

Imagen5:Captura de pantalla de nuestra home page

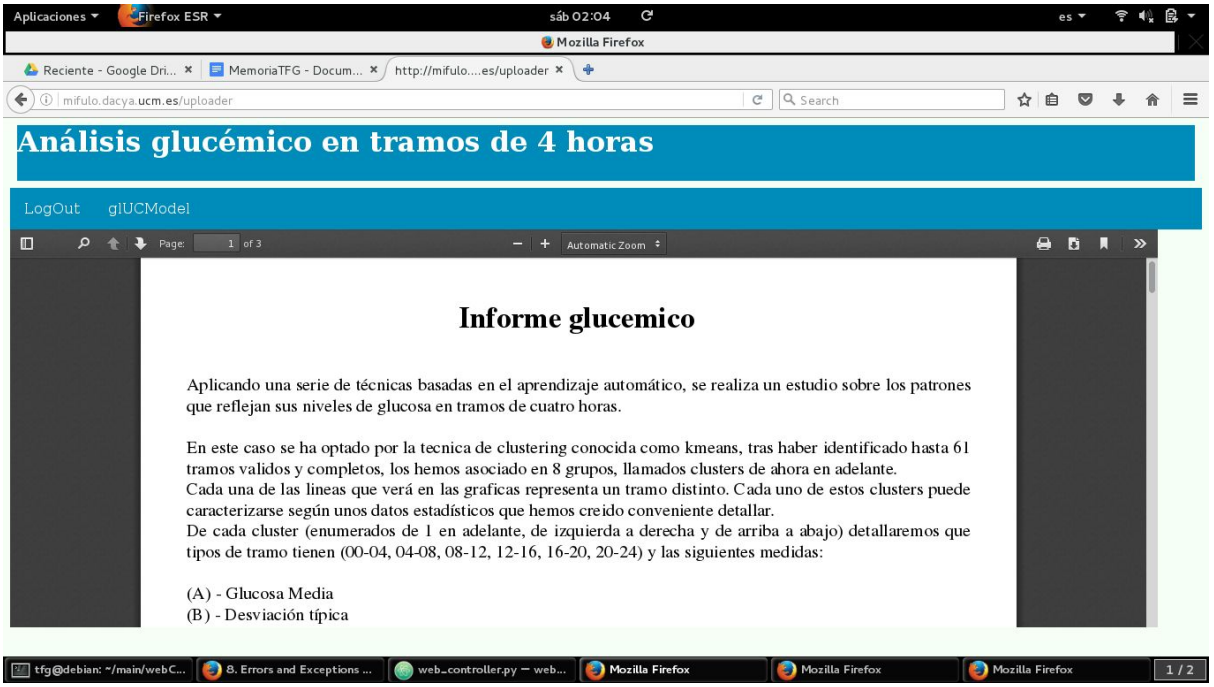

Imagen6:Captura de pantalla de nuestra página para mostrar informes

3.3. Desarrollo

Para explicar esta etapa, encontramos más sencillo, exponer de forma independiente el de la aplicación base y el de la aplicación web.

Evidentemente y tal y como se ha mencionado en el apartado referente al flujo del proyecto, la web llama al modelo (aplicación base).

3.3.1. Desarrollo de la aplicación base

Durante el desarrollo de la aplicación identificamos tres partes principales,

- Análisis y procesado de la entrada
- Aplicación de técnicas de aprendizaje automático sobre la entrada
- Recopilación y muestra de resultados

3.3.1.1. Análisis y procesado de la entrada

Como entrada, la aplicación recibe el archivo que emite el FreeStyle Libre, un fichero con extensión .txt, que tiene una estructura similar a un CSV(comma-separated values).
En este archivo el dispositivo citado ha ido guardando durante quince días una serie de registros en cada fila. Existen múltiples columnas que son rellenadas según la fila, desde niveles de glucosa, cantidad de insulina inyectada, ejercicio realizado…

Todas las filas rellenan las columnas de identificador, fecha,hora y minutos y la que indica el tipo de registro al que pertenece esa fila. Nosotrxs nos centraremos en los registros tipo 0, que cada quince minutos toman los valores de azúcar en sangre y son los que nos pueden dar información muy concisa del día a día del paciente.

En este momento tenemos los niveles de glucemia de quince días tomados cada quince minutos. Estos serán los datos con los que empezaremos a trabajar, agrupándolos cada cuatro horas y tratando de mantener un identificador que nos permita, aún sin saber a qué día en concreto pertenecen, determinar en qué periodo del día se obtuvieron estos niveles.

Por problemas del hardware generador de datos algunos de estos tramos están incompletos o ausentes. Aun así más del 80% de la información nos sigue siendo útil.

3.3.1.2. Clasificación o agrupación de los tramos identificados

Para agrupar o clasificar los tramos que queremos estudiar existen dos opciones generales, los algoritmos jerárquicos y los no jerárquicos.

Los algoritmos jerárquicos, como HBDSCAN consisten en ir asociando datos que comparten similitud en los atributos que los forman, e ir juntando esas asociaciones hasta que se llega al conjunto global.

El número de clusters que se obtendrán se basará en unas puntuaciones, como la Calinski-Harabasz que describen la homogeneidad de un grupo y el grado de diferencia con el resto. Además de este valor también se tiene en cuenta parámetros de entrada como elementos mínimos para formar clusters, separación máxima entre elementos…

No todos los algoritmos jerárquicos son aglomerativos, algunos son disociativos y parten del todo para ir haciendo grupos más pequeños, al contrario de lo que se acaba de describir.

Los algoritmos no jerárquicos, como KMeans o Agglomerative Clustering, tienen una estructura similar entre ellos y distinta a los jerárquicos, la diferencia se basa en que estos tienen prefijado el número de clusters que se va a obtener.

Esto puede llevar a tener clusters que no tengan mucho sentido si no se escoge con cuidado el número de ellos. Para esto se va probando el método con distintos números de clusters y se escoge el que resulte más apropiado.

Existen varios métodos para tal propósito, por su facil implementación en este caso se cogío una metrica parecida a la descrita anteriormente llamada Silhouette, la cual se recomienda en este tipo de algoritmos.

Se tomó la decisión de probar tres algoritmos, HBDSCAN, KMeans y agglomerative.

Se empezó trabajando con KMeans y ocurrió un problema al principio (Figura 1). Al sencillamente al calcular la distancia euclídea para decidir los grupos, se asociaban tramos que no tenían que ver entre sí más que la cercanía de los valores en las posiciones que los formaban, es decir no se tenía en cuenta la pendiente, primando la distancia entre los valores a las tendencias de las rectas.

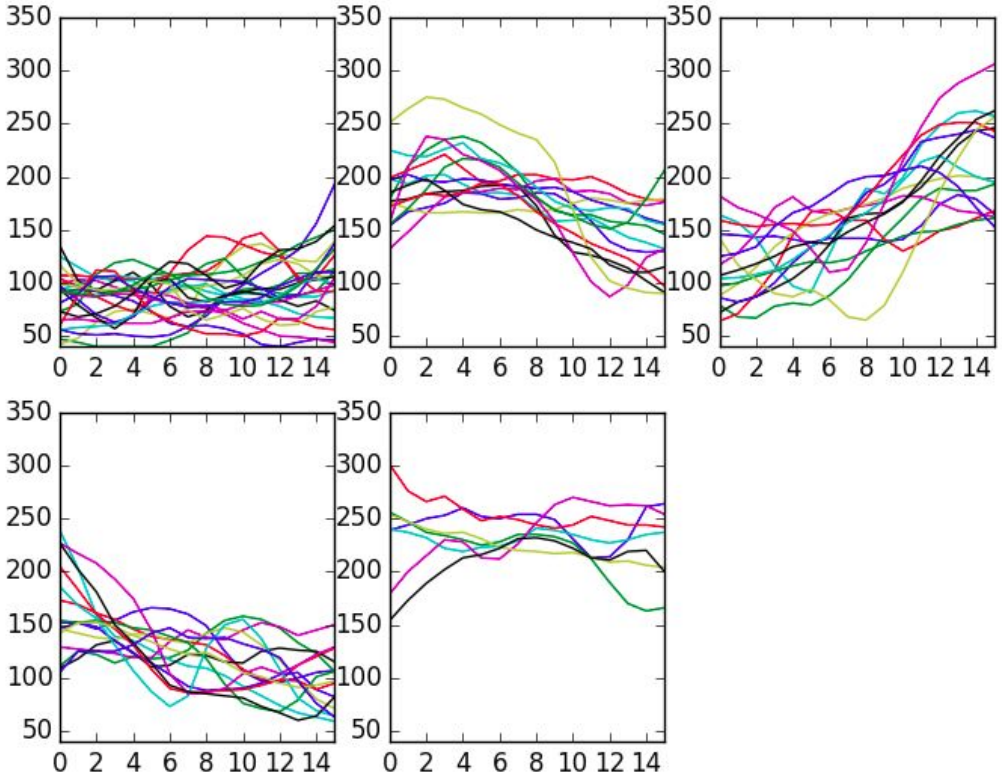

Figura1:Método KMeans, distancia euclídea, sin normalizar

Para solucionar esto, en vez de usar los tramos como entrada de los algoritmos de clasificación se usaron los resultados de la función de normalización (figura 2).

Además de utilizar la métrica Silhoeutte, también nos ha parecido interesante permitir decidir de forma manual el número de clusters que se genera.

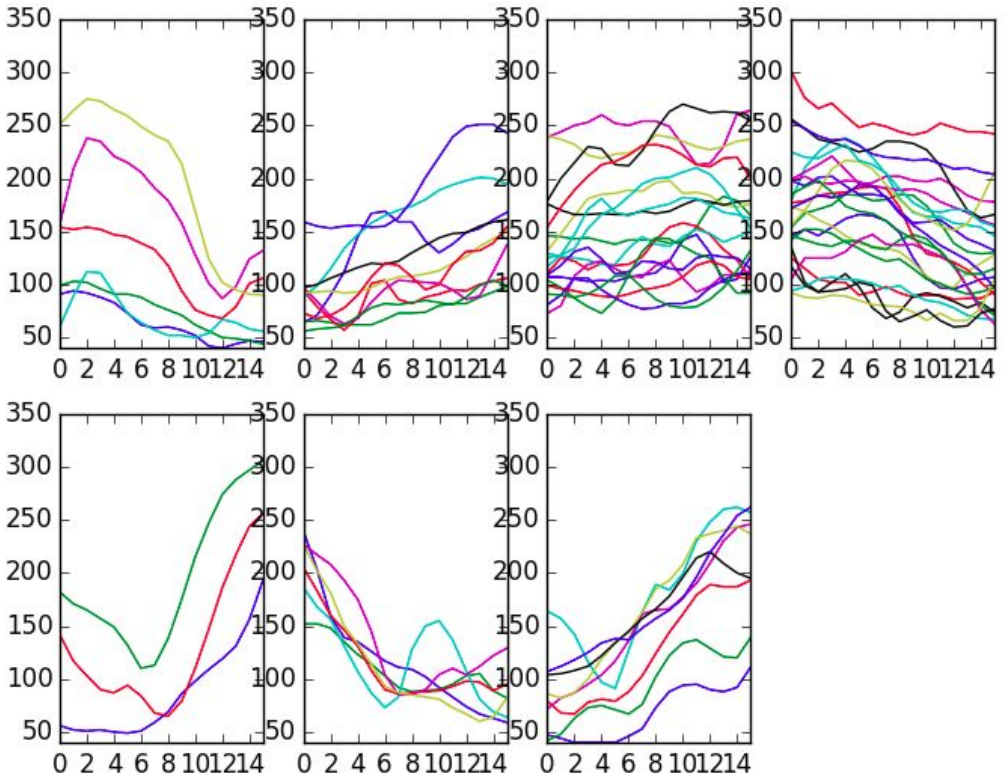

Figura2:Kmeans, distancia euclídea, con normalización de funciones

Los resultados con agglomerative tras la normalización gozan también de bastante interés.

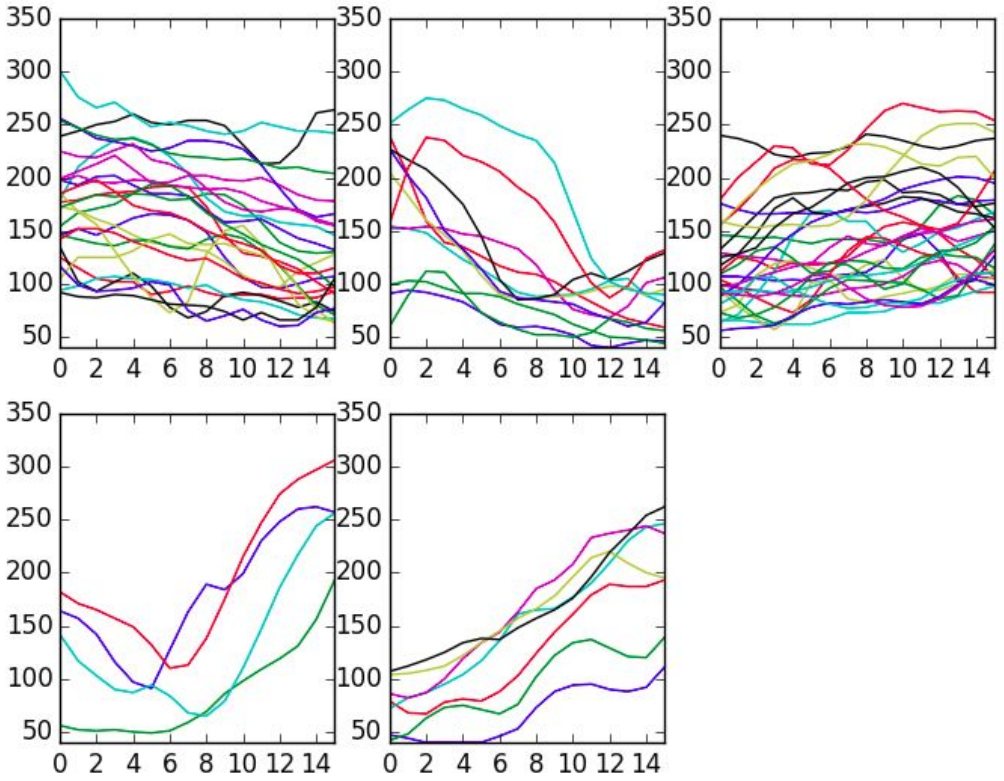

Figura 3:Agglomerative normalizada con métrica Manhattan

Con HDBSCAN no se han obtenido resultados positivos, ya que la mayor parte de los cluster solo tenían dos tramos, existía demasiada correlación en cluster diferentes(figura 4) y si se pedían que los cluster fueran mayor de dos tramos más de la mitad de estos se consideraban ruido(figura 5). El último de los cluster corresponde a los tramos descartados.

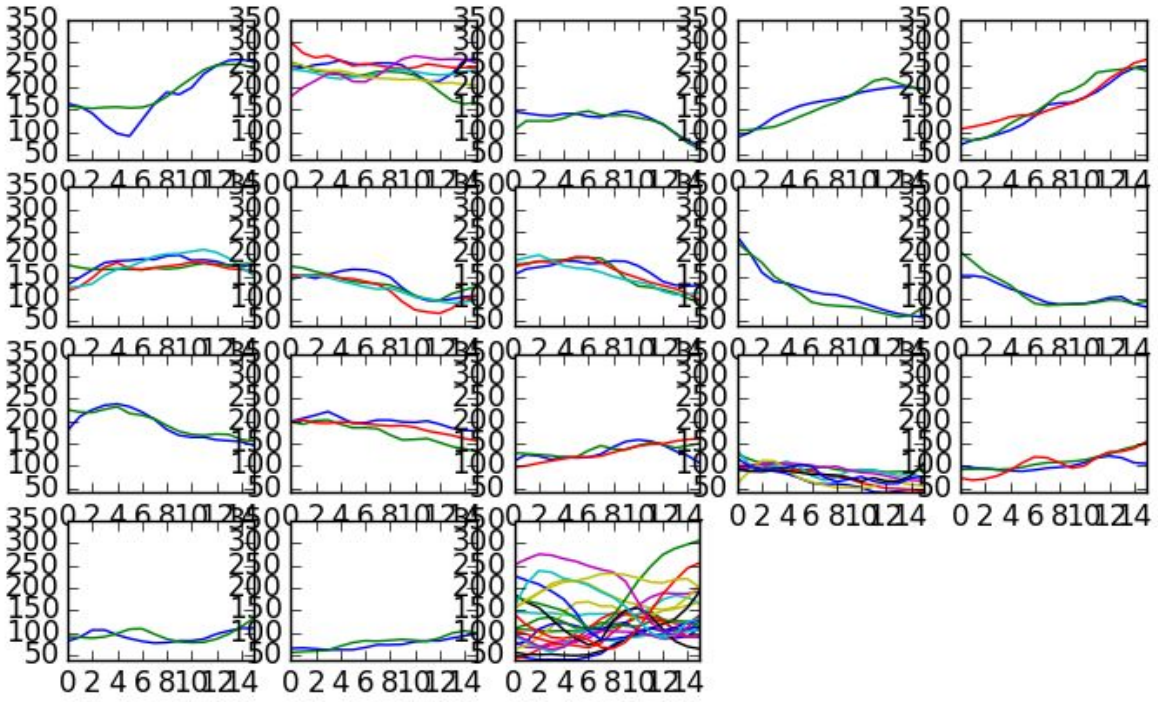

Figura4:HDBSCAN, métrica l2 normalización euclídea (min_cluster_size=2,min_samples=1)

El último cluster corresponde al ruido

3.3.1.3. Muestra de los resultados

La forma de mostrar esta información al usuario de la web será mediante un documento portable formal (PDF) en el que además de mostrar las gráficas se le explica a qué periodos del día corresponden los tramos de cada cluster.

Ejemplo:

El cluster número 7 está formado por 2 tramos correspondientes a las horas entre medianoche y las 4 am, 2 tramos correspondientes a las horas entre las 4 am y las 8 am, 3 tramos correspondientes a las horas entre las 4 pm y las 8 pm, 1 tramo correspondiente a las horas entre las 8 pm y medianoche.

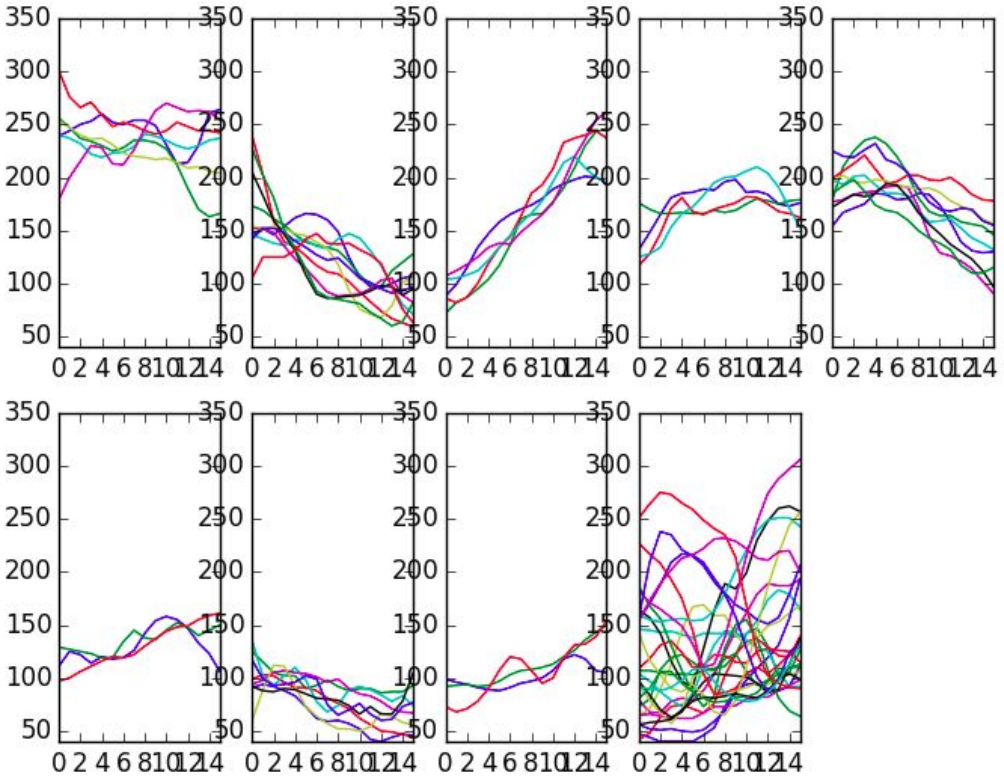

Figura5:HDBSCAN, métrica l2

(normalización euclídea)(min_cluster_size=3,min_samples=1)

El último cluster corresponde al ruido

Además se le muestra una tabla con unos datos estadísticos informativos de cada cluster:

(A) - Glucosa Media

(B) - Desviación típica

(C) - Glucosa máxima media

(D) - Glucosa mínima media

(E) - Porcentaje de tiempo en rango 70-180

(F) - Número medio de eventos por debajo del mínimo(60)

(G) - Número medio de eventos por encima del máximo (240)

(H) - Máximo número de eventos fuera de rango(60-240)

(I) - Mínimo número de eventos fuera de rango(60-240)

Al final del documento se advierte al usuario de que mantenga las pautas indicadas por su médico:

Los resultados de estos análisis son meramente experimentales, producto de la aplicación de técnicas de aprendizaje automático SIN SUPERVISIÓN MÉDICA con los datos que usted nos ha proporcionado. Bajo ninguna circunstancia sustituya las pautas de su médicx por las conclusiones que puedan resultar de esta lectura.

Ni la UCM, ni el grupo de investigación se harán responsables del uso indebido que pueda derivarse de este documento. Ante cualquier duda acuda a su endocrinólogx o médicx de cabecera.

El contenido de este documento ha sido generado de forma automática por el software MeanSerpentForest.

Además de esto se almacena el resultado de cada tramo en un archivo que contiene la lista de los dieciséis valores. Dicho fichero se llama un nombre que indica el periodo del día en una carpeta que se llama como el cluster al que pertenece.

Esto tiene como objetivo facilitar el estudio en futuras investigaciones que puedan utilizar estos datos como creyeran convenientes.

Cabe destacar que ha ocurrido un problema con los caracteres especiales (vocales con acentos) que había en el código, de esta forma el informe final carece de ellos en algunas partes, que no en todas.

3.3.2. Desarrollo de la aplicación web

La atención de peticiones, está gestionada por un script en python que utiliza una biblioteca WSGI llamada Bottle. Nos referiremos a este script como controlador a partir de ahora.

Este script, envía al cliente las distintas vistas (páginas HTML), que le muestra su navegador durante su interacción con nuestra página. Gracias al motor <<*SimpleTemplate (engine)*>>, podemos desacoplar los componentes de la web, llamando desde el controlador en python a métodos que referencian vistas (documentos HTML con extensión .tpl) sacando cualquier referencia a las mismas (HTML, CSS,JavaScript..) de aquel.

También es el encargado de recuperar los ficheros subidos, y los valores que seleccionados para los componentes visuales, por el usuario. Estos datos los extrae de los POST-request que recibe del cliente.

Nuestra “landing page” permite al usuario identificarse utilizando sus credenciales de Google gracias al proceso de autenticación delegada que se explica en el punto pertinente correspondiente a la seguridad del proyecto, en ese mismo documento.

Para llevar a cabo el proceso de autenticación, ha sido necesario registrarse como desarrollador en Google, dar de alta el proyecto, y habilitar el API de Google+. Y finalmente generar las claves para Oauth (client_id, secret, redirect_uri).

Para añadir estilo a las páginas generadas, utilizamos CSS (Cascading Style Sheets).

En cuanto a la integración con la aplicación base. Suponiendo que algún usuario, se ha identificado, y ha subido un documento con sus registros médicos. Se realiza una llamada al método de clustering que haya elegido el usuario y con tantos núcleos como haya seleccionado. Dicha llamada es posible gracias a que se importa la clase principal de la aplicación base como módulo.

3.4. Seguridad

Al tratarse de datos médicos, los ficheros que nos suben los pacientes deben gozar del mayor de los niveles de seguridad.

3.4.1. Restricciones legales

La política de seguridad aplicada tiene en cuenta la normativa relacionada con la seguridad.

En primer lugar se debe cumplir con lo dispuesto en la Ley Orgánica de Protección de Datos Personales (LOPD).

Esta Ley ha sido objeto de desarrollo reglamentario por el Real Decreto 1720/2007 . Según estas normas los datos personales que se tratan por la aplicación desarrollada tienen la consideración de datos personales especialmente protegidos. Las medidas de seguridad que se deben aplicar son las correspondientes a datos de nivel Alto.

Algunas de estas medidas son:

- Deberá cifrarse la información almacenada
- La transmisión de datos deberá realizarse de forma cifrada
- Se deberá registrar para cada acceso la identificación del usuario, el registro accedido y la fecha y hora del mismo.

Además establece con carácter general de confeccionar un documento de seguridad.

En segundo lugar debe mencionarse el Real Decreto 3/2010 por el que se regula el Esquema Nacional de Seguridad. Esta norma es aplicable a los servicios públicos y también a las Universidades Públicas.

Esta norma surge para cumplir con la Ley 11/2007 de acceso electrónico a los Servicios Públicos Establece la necesidad de regular reglamentariamente la política de seguridad en la utilización de medios electrónicos para acceder a los servicios públicos.

El Esquema Nacional de Seguridad establece entre otros los siguientes principios básicos de seguridad:

- Prevención frente a interrupciones o modificaciones fuera de control
- Garantizar la imposibilidad de acceso a personas no autorizadas.

Define la seguridad de las redes y de la información como la capacidad de resistir con un determinado nivel de confianza, los accidentes o acciones ilícitas o malintencionadas que comprometan la disponibilidad, autenticidad, integridad y confidencialidad de datos y aplicaciones.

Recientemente (27 de abril de 2016) se ha aprobado el Reglamento General de Protección de Datos que es un Reglamento Europeo. Esta norma persigue la protección de los datos

de las personas y dispone la aplicación de un conjunto de medidas en todos los Estados de la Unión Europea para posibilitar la libre circulación de datos en la misma.

Entre estas medidas destaca la obligación de contar con el consentimiento de las personas para cualquier tratamiento de sus datos.

3.4.2. Documento de seguridad

El Reglamento de desarrollo de la LOPD establece la obligación de que el responsable del fichero elabore un documento de seguridad que recogerá las medidas técnicas y organizativas en línea con la normativa de seguridad vigente que será de obligado cumplimiento para el personal con acceso a los sistemas de información.

El documento de seguridad debe contener entre otros contenidos los siguientes::

- Identificación de la organización
- Los ficheros que tiene la organización, su estructura, origen de los datos, tipos de datos y nivel de seguridad de los ficheros
- Las medidas de seguridad adoptadas
- Información y obligaciones del personal
- Procedimientos de notificación, gestión y respuesta ante incidencias

Se ha incorporado al Trabajo Fin de Grado un ejemplo de documento de seguridad para el proyecto desarrollado.. Se ha tomado como referencia el modelo de documento de seguridad que la Agencia Española de Protección de Datos pone a disposición de las organizaciones.

3.4.3. Personas responsables

Se debe distinguir por una parte la figura del responsable del fichero, y por otra, al encargado del fichero.

El responsable del fichero es según la LOPD es la entidad, persona u órgano administrativo que decide sobre la finalidad, el contenido y el uso del tratamiento de datos personales. En el caso de que el servidor que alberga los datos proporcionados por los pacientes fuése de la Universidad Complutense de Madrid, el responsable del fichero sería la Universidad debiendo designar a una persona como representante de la misma.

Entre otras, las obligaciones que tiene un responsable de ficheros son:

- Comunicar la información relativa al fichero para que la Agencia Española de Protección (AEPD) de datos realice su registro en el Registro de la AEPD.
- Informar a los titulares de los datos sobre la recogida de los mismos
- Obtener el consentimiento del interesado para realizar el tratamiento de sus datos
- Facilitar y garantizar el ejercicio de los derechos de oposición, al tratamiento, acceso, rectificación y cancelación.
- Asegurar que el tratamiento que se haga de los datos se ajuste a la finalidad para la que fueron obtenidos

El encargado del fichero es la persona física o jurídica que realice el tratamiento de sus datos por cuenta del responsable del fichero..

3.4.4. Política de seguridad

Toda política de seguridad consta de medidas preventivas, medidas de monitorización y de análisis de vulnerabilidades.

3.4.4.1. Medidas preventivas

Las medidas preventivas aplicadas en este trabajo son las siguientes:

- Autenticación delegada
- Cortafuegos:.IPtables.
- Cifrado
- Copias de seguridad
- Control de peticiones
- Inhabilitación de las conexiones ssh para el usuario root

Las medidas preventivas están orientadas a evitar que se produzcan ataques.

3.4.4.1.1. Autenticación delegada

Cómo se ha mencionado anteriormente, la autenticación delegada, supone una solución eficiente a la tediosidad que supone la correcta gestión de claves de usuario, y facilita el cumplimiento de las medidas exigidas por la LOPD.

El funcionamiento de este sistema de autenticación, puede explicarse en las siguientes etapas:

1. El cliente accede a nuestra página principal, que contiene un enlace al punto de autorización de Google. La URL de este punto se obtiene del documento de descubrimiento, este documento es un JSON con distintas configuraciones necesarias para autenticar. Este enlace contiene varios parámetros:
    a. client_id: para que Google (en este caso) sepa qué aplicación quiere solicitar autenticación
    b. redirect_uri: A la que redirigiremos al cliente si se autentica con éxito.
    c. scope: Expresa qué datos se solicitan del usuario

2. .El cliente pulsa el enlace que lleva a una página de Google, donde introduce sus credenciales y da permiso a la aplicación para acceder a los datos solicitados en el parámetro scope.
3. Esta etapa involucra los siguientes pasos:
    a. El usuario envía sus credenciales a Google.
    b. Google genera un código temporal AAAA e informa al navegador que tiene que redirigirse a redirect_uri con dicho código AAAA como parámetro.
    c. La aplicación web intercambia el código AAAA por los dos tokens: id_token y access_token.
    d. La aplicación extrae la información sobre el usuario id_token que es un objeto JSON firmado y cifrado con la clave privada de Google.
4. Desciframos los tokens en el propio servidor usando los certificados de Google.

Con este sistema, facilitamos a los potenciales usuarios el acceso a nuestra aplicación, no tienen que registrarse. Y acceden a través de una red social de confianza.

3.4.4.1.2. Cifrado

Tal y como ha sido expuesto en el apartado relativo a la normativa aplicable los datos a tratar exigen la adopción de medidas de protección de nivel alto entre las que se incluye el cifrado de la información tanto en las comunicaciones como en el almacenamiento.

El cifrado de las comunicaciones se basa en la utilización de TSL/SSL del puerto 443
SSL encripta las comunicaciones TCP. Por tanto se puede utilizar para cualquier aplicación que utilice TCP. En este trabajo se ha optado por HTTP a través del puerto 443 (HTTPS) y no el puerto 80.

Para ello ha sido necesario contar con un certificado en el servidor. La autoridad de certificación que ha proporcionado el certificado ha sido startcomca.

El formato del certificado utilizado es X-509 v3 y tiene activado el bit de encriptación en el campo Key Usage. Incluye la clave pública que necesita el navegador que utiliza el paciente para cifrar la clave simétrica que genera para cada conexión con el servidor donde se localiza la aplicación de este Trabajo Fin de Grado.

También se cifra toda la información almacenada.
Se utiliza el paquete gnupg. Utiliza claves simétricas que son almacenadas en disco duro en ficheros llamados llaveros. Existe un llavero para las claves públicas utilizadas y otro para las claves privadas.
El proceso de cifrado se basa en que la clave pública cifra la clave simétrica que se utiliza para cifrar la información. Sólo mediante la clave privada se puede recuperar la clave simétrica.

3.4.4.1.3. Iptables

El paquete cortafuegos es el producto IPTables. Este producto está contenido en la distribución Debian-Server de Linux. Este sistema operativo activa IPTables en el arranque ya que está directamente vinculado al kernel de Linux.

Se han añadido las reglas de filtrado apropiadas al proyecto. Estas reglas son las siguientes:

- Puertos origen del servicio
    - Ningún puerto bien conocido
- Puertos destino del servicio
    - 443 (HTTPS)
- Dirección IP origen. Se impide:
    - Dirección IP del equipo
    - Direcciones privadas:

        - 10.0.0.0 a 10.255.255.255
        - 172.16.0.0 a 172.31.255.255
        - 192.168.0.0 a 192.168.255.255
    - Direcciones IP multidifusión
        - 224.0.0.0 a 239.255.255.255
    - Direcciones reservadas de la clase E
        - 240.0.0.0 a 247.255.255.255
    - Direcciones de interfaz de bucle invertido
        - 127.0.0.0 a 127.255.255.255
    - Direcciones especiales
        - 0.0.0.0
- Dirección IP destino
    - la IP del equipo

3.4.4.1.4. Copia de seguridad

Una copia de seguridad (backup) es una copia de la información que se realiza como preventiva para el supuesto original se pierda o se dañe.

Las copias de seguridad pueden ser

- Completas: se copian todos los ficheros que se indique.
- Diferenciales: se copian únicamente los ficheros modificados después de la última copia completa
- Incrementales: se copian únicamente los ficheros modificados desde la última copia completa o diferencial.

La copia de seguridad sobre el equipo Debian utilizado en este TFG, se lleva a cabo utilizando la suite Mondo Rescue.

La frecuencia de la copia de seguridad prevista en el trabajo fin de grado es realizar una copia diaria.

3.4.4.1.5. Control de peticiones

Por el funcionamiento de la biblioteca que utilizamos para generar la web (Bottle.py), el único modo que tiene el usuario para acceder a las distintas páginas que alojamos en nuestro servidor, es recibiendolas como respuesta a las peticiones que le ofrecemos hacernos.

Si por ejemplo el usuario escribiese la url de cualquier página que no fuese la landing page, recibirá un error 405 (método no permitido).

3.4.4.1.6. Inhabilitación de conexiones ssh para el usuario root

Para posible proteger de intentos de acceso externos un nuestro servidor, hemos bloqueado el acceso al usuario root.

Realizando los accesos con cualquier otro usuario, y una vez dentro adquirir privilegios si fuera necesario (mediante sudo por ejemplo).

3.4.4.2. Medidas de monitorización

Las medidas de monitorización tienen como objetivo detectar incumplimientos de la política de seguridad y también de la normativa que regula la protección de datos personales.

La normativa española exige para los datos de nivel alto, el registro de los accesos almacenando la identificación del usuario, la fecha y hora en que se realizó el acceso, el fichero accedido y si ha sido autorizado o denegado.

No se ha incluido en el trabajo realizado la implantación de un sistema de intrusiones ya que se considera que son suficientes las medidas preventivas aplicadas.

3.4.4.2.1. Registro de acceso

El registro de los accesos se ha resuelto a un doble nivel:

- A nivel de sistema operativo para un filesystem mediante journaling
- A nivel de la infraestructura de aplicación utilizando la funcionalidad de CherryPy

Tal y como ha sido indicado el sistema de archivos utilizado es un sistema de archivos extendido transaccional ext4 que puede tener asociado un registro de transacciones que describe todas las operaciones del sistema de archivos en orden secuencial.

El sistema ext4 se ha configurado para mantener un registro de todos los cambios que se produzcan en el filesystem que almacena los datos proporcionados por los pacientes.
Para ello se activa la opción de journaling en modo journal-data.

Con ello se facilita, además, la recuperación de la información en caso de algún incidente que pueda ocasionar inconsistencias. Esta opción también cubre los accesos que haga el administrador del sistema.

Por otra parte se activa a nivel de infraestructura de CherryPy el log de accesos (log.acces_file). En esta esta opción se indica el nombre del fichero cuyos accesos se desea registrar.

Este fichero de log se guarda en un filesystem distinto que tiene permisos sólo de lectura.
La utilización de ambas opciones, journal y el log de accesos, permitirá responder a cualquier auditoría de accesos que se pueda plantear.

3.4.4.3. Análisis de vulnerabilidades

Vulnerabilidad significa debilidad. El análisis de vulnerabilidades pretende descubrir estos agujeros de seguridad antes que sean utilizados indebidamente.

Se ha previsto la utilización de un scanner de vulnerabilidades. Se ha optado por Nessus Vulnerability Scanner (v5.2).

Algunos de los tipos de vulnerabilidad que detecta son:

- Agujeros que permitan a un hacker acceder a la administración del sistema.
- Debilidades que faciliten un ataque de denegación de servicio.
- Detección de passwords de fácil suplantación.
- Falta de actualización de parches en el sistema operativo.

4. Resultados

Para concluir ofrecemos y analizamos algunos de los resultados obtenidos con nuestra herramienta.

4.1. Ejemplo de ejecución

El paciente fiat_miguel@yahoo.es ha subido un fichero en el instante 1497089000.07 desde 1970, ha seleccionado el método KMeans y el número 7 para los clusters, o la opción por defecto que en este caso son dos opciones indistinguibles.

Es un fichero que también incluye información sobre las inyecciones de insulina e ingesta de carbohidratos, esta información no será utilizada en la elaboración del informe. En este se empieza informando al usuario del análisis que se va a realizar, en el caso de haber seleccionado HDBSCAN se informaría de que el último cluster corresponde al ruido.
De manera que el paciente recibe un informe anónimo con las gráficas referentes a sus clusters, una pequeña explicación del origen temporal de los tramos y la información en forma de tabla y demás relatada en el punto 3.3.1.3.

En la carpeta asignada al efecto de almacenar la información a esta subida se guarda además del csv con la información del paciente, el fichero informe.pdf mostrado en la web, la gráfica llamada img.png, la ip de subida guardada como ip.txt y las carpetas nombradas con el nombre de los clusters que guardan los tramos que les corresponden.
El fichero que se genera es el informe adjunto.

4.2. Discusión crítica y conclusiones

Como requisito de este trabajo, desarrollamos este apartado tanto en castellano como en inglés.

4.2.1. Castellano

La página web ofrece un servicio a los pacientes que, con ayuda médica, puede ayudar a mejorar su calidad de vida. Con la falta de conocimientos médicos que tenemos no podemos llegar a comprender la verdadera utilidad de MeanSerpentForest, el software desarrollado.

Basándonos en el origen respecto a la hora del dia de los tramos que componen los distintos clusters. Este polimorfismo requiere de la intervención de un médico para comprender si estos patrones pueden desarrollar ideas que afecten a las indicaciones sobre los pacientes.

La idea original es buena y se puede seguir tirando de este hilo mediante la exploración del tema con algoritmos jerárquicos del tipo RandomForest. Si finalmente no se consigue una mejora en la calidad de vida de los pacientes con este proyecto esperamos que al menos sirva para recabar datos que sirvan a otras investigaciones a lograr estos objetivos.

4.2.2. English

The web site offers a service to the patients that, combined with medical advice, can help to improve their quality of life. Given our lack of medical knowledge, we can't actually understand MenSerpentForest's (the software we have developed) utility.

Based on the origin respect from the time of the day of the slots that compound different clusters. This polymorphism requires of the intervention of a doctor to understand whether those patterns can develop ideas that affect the indications about patients.

The original idea might work, and it would be possible to go deeper into it by the exploration of the topic with hierarchical algorithms such as RandomForest. If we finally didn't reach the goal of improving patients' quality of life with this project, we would hope, at least, that this work will be useful for future and related investigations.

Anexo I: Manual de uso.

Existen dos formas de usar la aplicación, una es la explicada anteriormente a través de la web. También se puede utilizar la aplicación llamándola desde la terminal, en este caso no se produce el PDF ni ninguna gráfica sino sólo la separación de tramos en clusters en carpetas en la ruta indicada.

Este modo de uso no es para pacientes sino que se restringe a la investigación

La llamada a python se realizaría de la siguiente manera:
python src.py nombre_de_ruta metodo_clustering num_clusters
nombre_de_ruta es la ruta que contiene el fichero a analizar. Es decir, tiene que acabar, con el carácter "/". Dentro de esta carpeta es recomendable que solo exista un fichero. Debe llamarse "csv.txt" y ha de ser el que contenga los registros para la entrada.

Si no se indica el número de clusters se calcula en función de la métrica correspondiente, lo cual es recomendable, si no se indica el algoritmo por defecto se usa KMeans.
Si no se indican argumentos o el primero es *help* se imprime esta información por el terminal.

Anexo II:Documentación y asesoramiento.

En este anexo incluimos la bibliografía (libros y enlaces) en la que nos hemos apoyado tanto para desarrollar el proyecto, como para redactar éste y los distintos documentos a los que nos referimos en el Anexo IV.

También mencionamos a las personas que han contribuido de forma completamente altruista, a que tanto el software como la documentación que hemos producido, cumpla en la medida que nos ha sido posible los criterios de programación eficiente, y los requisitos legales de los que hemos sido conocedores.

A. Bibliografía.

- Seguridad Informática, Gema Escrivá, Rosa Mª Romero, David Jorge Ramada y Ramón Onrubia Pérez.
- Redes de computadoras Un enfoque descendente, James F.Kurose Keith W.Ross
- Firewalls Linux, Robert L. Ziegler
- Firewall, La seguridad de la banda ancha, José Antonio Carballar
- Inteligencia Artificial, Técnicas, métodos y aplicaciones, Edición coordinada por José T.Palma Méndez y Roque Marín Morales; Técnicas de agrupamiento desarrollado por Amparo Vila Miranda y Miguel Delgado Calvo-Flores
- Tabla de alimentos para diabeticos, Doris Fritzsche
- Introducing Python, Modern Computing in simple packages, Bil Lubanovic
- Foundations of Python Network Programing, Brandon Rhodes and John Goertzen
- Protocolos Criptográficos y Seguridad en Redes, Jaime Gutiérrez y Juan Tena
- Python Playground, Geeky Projects for the Curious Programer, Mahesh Venkitachalam
- Network Security with OpenSSL John Viega, Matt Messier, Pravir Chandra
- El gran libro de Debian GNU/Linux, Rafael Eduardo Rumbos Salomón
- Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar

B. Enlaces de interés.

- Documentación oficial sobre autenticación con Google: https://developers.google.com/identity/protocols/OpenIDConnect
- Explicaciones paso a paso sobre las fases de la autenticación con Google(rutas,peticiones y parámetros involucrados): https://developers.google.com/identity/protocols/OpenIDConnect#server-flow
- Consola del desarrollador de Google: https://console.developers.google.com
- Añadiendo estilo CSS: http://pythonhow.com/add-css-to-flask-website/
- Bottle: http://bottlepy.org/docs/dev/
- Enrutando archivos estáticos: http://bottlepy.org/docs/dev/tutorial.html#routing-static-files
- API sklearn: http://scikit-learn.org/stable/modules/classes.html
- API HDBSCAN: http://hdbscan.readthedocs.io/en/latest/api.html
- Habilitar y deshabilitar listas descendentes: https://www.w3schools.com/js/tryit.asp?filename=try_dom_select_disabled
- Habilitar y deshabilitar selector en función de otro selector: https://stackoverflow.com/questions/12269470/selectbox-disabled-or-enabled-by-an-option-of-an-other-selectbox
- Survey of Clustering Data Mining Techniques Pavel Berkhin: www.cc.gatech.edu/~isbell/classes/reading/papers/berkhin02survey.pdf
- Autoridad de certificación que proporciona certificados: https://www.startssl.com
- Documentación sobre el sensor FreeStyle:

https://freestylediabetes.co.uk

http://www.freestylelibrepro.us/blood-glucose-monitoring-device.html

- Comparativa entre Dexcom G4 y FreeStyle Libre: www.jediazucarado.com/dexcom-g4-vs-freestyle-libre
- Modelo de monitorización glUCModel; J.Ignacio Hidalgo, Esther Maqueda, José Luis Risco-Martín, Alfredo Cuesta-Infante, J.Manuel Colmenar, Javier Nobel: https://www.ncbi.nlm.nih.gov/pubmed/24407050
- Tecnología en Diabetes. Grupo de Bioingeniería y Telemedicina. UPM: www.gbt.tfo.upm.es/Diabetes+technology
- 2008-2013 Action Plan for the Global Strategy for the Prevention and Control of Noncommunicable Diseases World Health Organization: www.who.int/nmh/publications/9789241597418/en
- Situación de la diabetes en España: http://www.fundaciondiabetes.org/prensa/297/la-diabetes-en-espana
- enlaceTabla1: https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/guia_de_seguridad_en_servicios_dns.pdf
- enlaceLibreLink https://www.librelink.com/es
- enlaceSiDiary https://www.sidiary.org/

C. Agradecimientos.

A parte de nuestros director y codirector de proyecto, y aprovechando nuestra condición de estudiantes de último curso, hemos recurrido a otros profesores de nuestra facultad a pedir asesoramiento o consultar dudas, de temas muy concretos y que estuvieran relacionados con alguna asignatura de la que nos hubieran dado clase a alguno de los dos.

Merece mención el hecho de que todos ellos, siempre que se lo hemos pedido, nos han respondido con diligencia, rapidez y eficacia. Es por ello que quisiéramos agradecer su contribución a este proyecto a los siguientes profesores:

- Enrique Martín Martín
- Inmaculada Pardines Lence

Además de a los profesores citados, también se ha consultado al responsable en seguridad de la AEAT:

- Alberto Zapico,

a quien también quisiéramos agradecer su dedicación, así como a los doctores:

- Iñigo Segarra Sánchez-Cutilla,
- Mª Angeles Raquejo Grado y
- Riánsares López Palomar.

Por último nos gustaría mencionar a nuestros compañeros de la asignatura de DSI, que no pusieron ningún tipo de problema cuando propusimos que la temática del trabajo TrabajoDSI, estuviera directamente relacionado con nuestro trabajo final de grado, nuestro grupo aparte de los dos autores de este documento, estaba compuesto por los alumnos de la Facultad de Informática:

- Adrián Martínez Jiménez
- Daniel Gutierrez Delgado
- Roberto Díaz Gómez

Anexo III: Enlaces del proyecto.

Enlace en github a software modelo:

https://github.com/proyectoNinja/meanSerpentForestSRC

Enlace en github a documento de seguridad:

https://github.com/proyectoNinja/security_docs

Enlace en github al código web:

https://github.com/proyectoNinja/webConBottle

Enlace a informe ejemplo:

https://github.com/proyectoNinja/meanSerpentForestSRC/blob/master/informe.pdf

URL aplicación web:

http://mifulo.dacya.ucm.es/

Anexo IV: Referencias.

Se muestra a continuación, el conjunto de tablas, imágenes y gráficos que mencionamos, o que aparecen en este documento. En los casos que procede, dichos elementos están enlazados en el apartado de documentación

Tabla1	Principales indicadores de la diabetes	enlaceTabla1
TrabajoDSI	Trabajo realizado para asignatura de interfaces durante el 1º cuatrimestre	enlaceComps
Imagen1	Capturas de pantalla de la aplicación Libre Link	enlaceLibreLink
Imagen2	Capturas de pantalla de la aplicación SiDiary Diabetes Management	enlaceSiDiary
Figura1	MeanSerpentForest - Kmeans distancia euclídea	MSF
Figura2	MeanSerpentForest - KMeans distancia L2	MSF
Figura3	MeanSerpentForest - Agglomerative distancia L1	MSF
Figura4	MeanSerpentForest - HDBSCAN min_cluster_size=2	MSF
Figura5	MeanSerpentForest - HDBSCAN min_cluster_size=3	MSF
Imagen3	Captura de pantalla de nuestra landing page	url_mifulo
Imagen4	Captura de pantalla de nuestra página de error	url_mifulo
Imagen5	Captura de pantalla de la página que permite al usuario subir ficheros	url_mifulo
Imagen6	Captura de pantalla de la página con la que mostramos el pdf generado	url_mifulo

Anexo V: Contribuciones

A. Miguel Fuentes

Durante el curso 2015/2016 me matriculé en la asignatura "Minería de datos y el paradigma Big Data". Gracias a ello me familiaricé con el entorno de Python y en concreto con el módulo de sklearn.

El empeño que ha puesto Lucas en desarrollar la aplicación web me ha permitido poder dedicarle mucho más tiempo al estudio y desarrollo de este nuevo mundo que ha sido para mi el aprendizaje automático.

Inicialmente completé las explicaciones de los tutores del trabajo de fin de grado sobre el tema de la diabetes con una entrevista con una médica, Riánsares López Palomar.

Una vez se nos asignó el puesto de trabajo, hice una puesta a punto inicial al mismo, pese a ser un ordenador de bajos recursos, instalé Debian porque era lo que Lucas y yo conocíamos. Instalé el servidor ssh, sudo, pip y atom para el desarrollo del código Python.

A los tres días el ordenador dejó de arrancar y tuve que volver a hacerlo.

En lo que respecta a la implementación del código de MeanSerpentForest soy el máximo responsable. Habiendo llevado un registro de los enlaces que visitaba, incluso de cada línea que he escrito y borrado y de las funciones que he implementado aunque finalmente no fueran necesarias. Estos archivos se llaman "enlacesDeInteres", "papeleraDeCodigo" y "moduloInutil.py" y están disponibles en github.

También me he hecho cargo de la generación del documento PDF, desde la distribución de las gráficas, la selección de módulos para su creación automática con su correspondiente código y de escribir los textos que los acompañan.

Desde el momento en el que produje el primer prototipo de mi programa, referido anteriormente como aplicación base o modelo, me he encargado de verificar y corregir en su caso la funcionalidad del mismo. Así como a detallarle a mi compañero el escenario y los parámetros con los que tenía que invocar al módulo principal que he definido.

Tanto para el desarrollo de todo el código que he diseñado, como para la redacción de todos los documentos relacionados con este proyecto, incluyendo esta memoria, he puesto en práctica, en la medida que me ha sido posible, todos los principios de programación que he adquirido durante esta carrera.

Además he intentado respetar, en la medida en la que he entendido que no perjudicaba al proyecto y que no chocaban con los de mi compañero, mis propios valores.

Es por ello que tanto Lucas como yo, solo hemos utilizado software libre en este proyecto. Además he tratado de sustituir en los morfemas de género la o/a por x en toda parte de la memoria en la que he participado con objetivo de visibilizar el problema que construye esta sociedad en torno al género.

He colaborado con Lucas en el diseño y toma de decisiones que hubo sobre el servidor y el sistema de ficheros así como un curso on-line de Angular2 y NodeJS que realizamos para tratar de tener una página web que no pareciera un mercado de la DeepWeb.

Le he dado a este trabajo ciento veinte holgadas horas de mi vida como si de un ritual satánico se tratara.

B. Lucas Segarra

Durante el primer cuatrimestre dediqué cierto tiempo a investigar tecnologías web, ya que en un primer momento desconocía que una parte de este proyecto consistía en desarrollar una web que permita a potenciales usuarios interactuar con los métodos de clustering que ha desarrollado Miguel.

Afortunadamente, durante la primera parte del año, cursé la asignatura de Gestión de Información en la Web, en la que aprendí a utilizar distintos frameworks para Python, que facilitan el desarrollo de aplicaciones web. De entre todas ellas, encontré en Bottle una forma sencilla de gestionar peticiones HTTP y devolver al navegador del cliente páginas en HTML.

Para desarrollar la web de este trabajo, he tenido además que familiarizarme con algunas herramientas web como CSS y JavaScript.

También al principio del curso, desarrolle un módulo en Python con una función que permite parsear documentos csv, con el mismo formato con el que los genera FreeStyle Libre. Posteriormente, Miguel decidió no utilizar dicha función para tratar los ficheros que recibimos, por existir múltiples bibliotecas en Python que permiten hacerlo de forma eficiente.

Por lo aprendido en la asignatura de GIW, y también en la de Seguridad En Redes, he asumido las labores de seguridad informática.

Tras consultarlo con los profesores de ambas asignaturas, y tal y como se ha indicado, entendí que utilizar autenticación delegada, supone una solución robusta y eficiente para el siempre tedioso proceso de identificación.

Así mismo configure las reglas de las iptables, intentando dotar de la mayor seguridad posible a nuestro sistema.

También he tenido reuniones de asesoramiento con profesionales en el campo de la seguridad informática y en la medicina. Así como con profesores de la Universidad. Estas personas están mencionadas en el campo de agradecimientos.

Durante esas reuniones, se me hizo saber la necesidad de redactar un documento de seguridad, que se adjunta con esta memoria. Con respecto a este documento quisiera mencionar que entiendo que el responsable del mismo es la Universidad, y que si se llegase a utilizar por pacientes reales, sería responsabilidad de la misma nombrar a un encargado que completase el documento que he redactado, y llevase a cabo las medidas que en él se detallen.

Para poder utilizar HTTPS, aparte de necesitar otro framework de Python que es compatible con Bottle, llamado Cherry, he tenido que emitir el certificado de nuestro servidor así como generar una clave pública asociada a dicho certificado. También almacenamos el certificado de la autoridad que nos certifica a nosotros.

Para agilizar las conexiones por SSH, o permitir a Miguel reiniciar la aplicación web sin que tenga que saber cómo se llaman, o donde están guardadas mis ficheros de código, he escrito scripts en Linux que realizan de forma robusta tales tareas. Para permitir reiniciar el proceso de escucha por SSH, y que no se detenga al cerrar la conexión, he utilizado la biblioteca nohup.

Llegada la última mitad del curso, y teniendo Miguel bastante completa su parte, mi labor ha consistido principalmente en integrar junto con mi compañero el código que cada uno habíamos generado de forma independiente, y realizar las pequeñas modificaciones de funcionalidad en la web que me ha indicado.

Aparte de lo que acabo de exponer, quisiera mencionar que tanto las decisiones de diseño, como la configuración del servidor que aloja nuestra aplicación, se realizaron de forma conjunta, y que todas las decisiones de desarrollo que he tomado, las he consensuado con mi compañero.

Con respecto a esta memoria, soy el responsable de la estructura que tiene, así como de su corrección léxica y ortográfica. También de la traducción al inglés de las partes pertinentes.

En cuanto al contenido de la misma, cada uno hemos completado los apartados de los que nos hemos encargado y revisado lo que ha escrito el otro.